

Chapter 17

ENVISIONING (PRODUCT PLANNING)

Before beginning the first customer-value-creation sprint, we need an initial product backlog. And to generate an initial product backlog we need a product vision. Many organizations also find it useful to create a preliminary product roadmap, which defines a potential series of incremental releases. Your organization might have other front-end artifacts that it prefers to create as well. I refer to the activity of creating these artifacts as envisioning, or product-level planning.

In this chapter I describe an envisioning approach that is well aligned with Scrum principles. It is also very useful for organizations that are trying to develop products using Scrum but must still integrate with a front-end approval process that is not agile.

Overview

Let's say you have an intriguing idea for a new product or the next version of an existing product. The goal of envisioning is to expound upon that idea, describing the essence of the potential product and creating a rough plan for how to approach its creation. At the end of envisioning, you should have sufficient confidence to subject the idea to portfolio planning (see Chapter 16), where you can decide whether you want to fund the next level of more detailed development.

Envisioning a product in Scrum should not be confused with heavier-weight, ceremonial, plan-intensive project chartering. With Scrum, we don't believe that we can (or should try to) know all the details about a product before we start. We do, however, understand that product funding usually can't move forward without first having a vision; enough details to understand the customers, features, and high-level solution; and an idea of how much the product might cost.

We don't spend too much time or effort envisioning because we want to quickly advance past the guessing stage, where we *think* we know the needs of the customer and the potential solution, to the fast-feedback stage—the customer-value-creation sprints. After all, it's only when we actually start implementing the solution through a continuous cycle of interactions with our complex environment that we acquire validated learning based on the reality in which our product must exist and thrive.

Timing

Envisioning, which is concerned with product-level planning, is an ongoing activity, not a one-time event (see Figure 17.1).

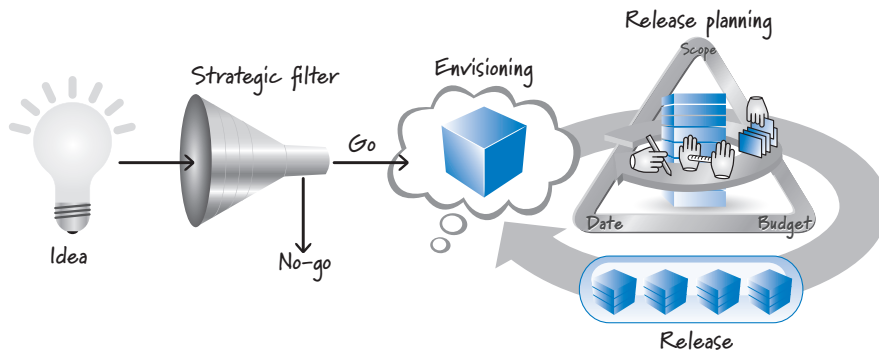


FIGURE 17.1 Envisioning is an ongoing activity.

Envisioning begins with an idea for a product that someone or some team has generated (a process often referred to as ideation). This idea is first passed through the organization's **strategic filter** to determine if it is consistent with the organization's strategic direction and therefore worthy of deeper investigation and investment.

Once the idea has cleared the strategic filter, we do initial envisioning. During this process, we generate enough understanding of the desired future product to define what the minimal first release should be. Doing so allows us to deliver high value quickly and at a low cost. It also puts something tangible in the hands of the actual users and customers as soon as possible, giving us actionable feedback to confirm or refute the assumptions we made about the target customers, the desired set of features, and our overall solution. This feedback might be in line with our expectations, reinforcing our desire to persevere with our current vision. On the other hand, it could just as easily be completely different from what we expected, which would cause us to pivot from our original solution, reenvision what we are doing, and modify the plan accordingly.

Participants

The product owner is the only required participant during initial envisioning. Normally, though, the product owner oversees an initial envisioning that includes one or more internal stakeholders, who collaborate with the product owner to perform the envisioning work. In addition, specialists in areas such as market research, business-case development, user-experience design, and systems architecture frequently participate in various envisioning tasks as well. Figure 17.2 illustrates the envisioning activity (optional participants and artifacts are indicated with dashed outlines).

Ideally, the ScrumMaster and the development team that will be performing the customer-value-creation sprints will also participate in initial envisioning, lending

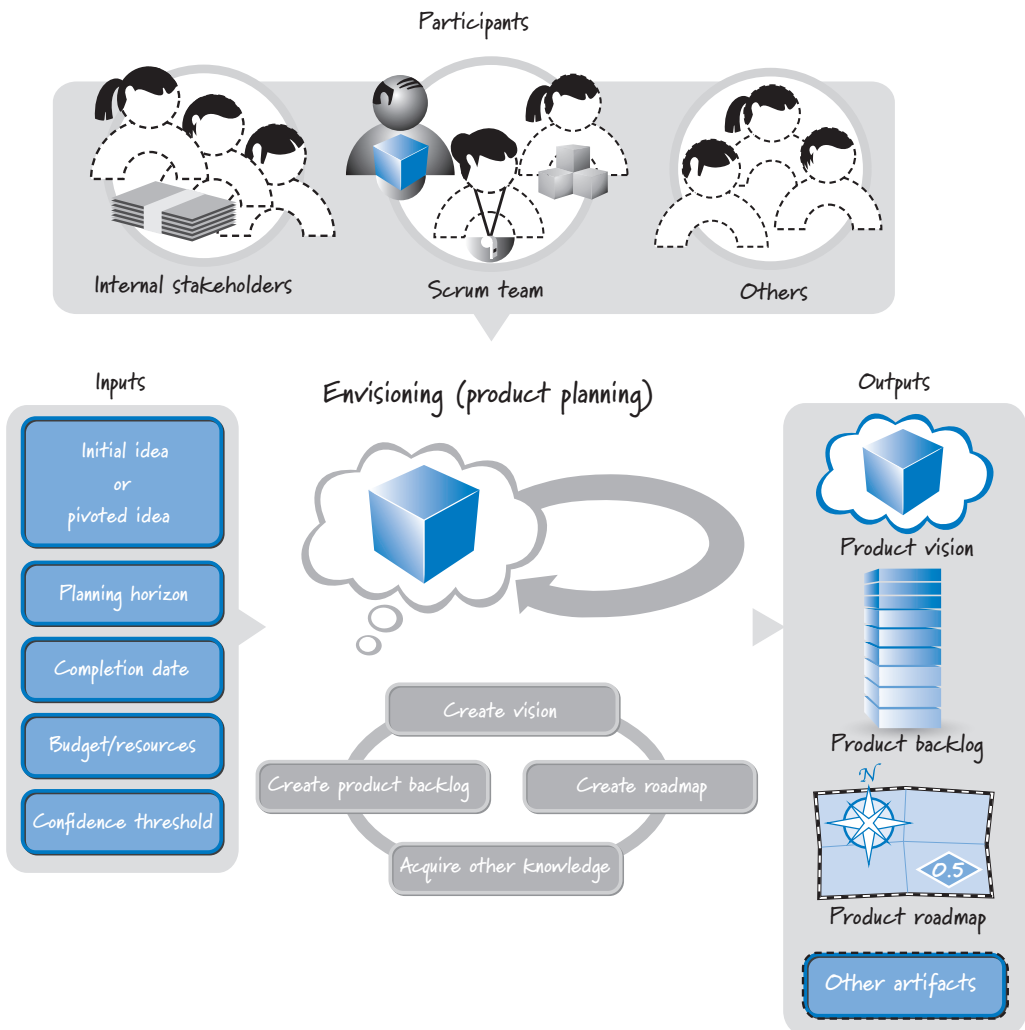


FIGURE 17.2 Envisioning (product-planning) activity

valuable feedback to the product vision and also eliminating the need to hand off the vision to another team to build the product. Often, however, the organization waits until initial envisioning is complete to fund the Scrum team, making it impossible to include it in initial envisioning activities. Once product development is under way and the full Scrum team has been allocated, however, the full Scrum team (product owner, ScrumMaster, and development team) should be included in any reenvisioning efforts.

Process

The main input for initial envisioning is an idea that has cleared the strategic filter. The main input for reenvisioning, on the other hand, would be a pivoted idea. Such an idea is one that has been updated or revised based on user or customer feedback, funding changes, unpredictable moves by competitors, or other important changes that occur within the complex environment in which ideas must exist.

We need other inputs as well. First, we need an indication of the planning horizon—how far into the future we should consider when we envision. We also need to know the expected completion date for the envisioning activities (if there is one), and the quantity and type of resources available to conduct envisioning. Last, we need to know the **confidence threshold**—the “definition of done” for envisioning, if you will. The confidence threshold is the set of information that the decision makers need in order to have enough confidence to make a go/no-go funding decision for more detailed development. I will talk more about what constitutes a reasonable confidence threshold later in the chapter. Finally, all of the envisioning inputs in Figure 17.2 should be considered simultaneously, not linearly.

Envisioning itself is composed of several different activities, each generating an important output such as the product vision or the initial product backlog. Frequently a simple product roadmap illustrating the incremental set of near-term releases is created as well. During envisioning, we can also perform any other activities that help us achieve the targeted confidence threshold in an economically sensible way.

SR4U Example

To illustrate the envisioning activity I use a fictitious new product idea called Smart-Review4You (or simply SR4U). The company, Review Everything, Inc., is a leader in online, consumer-supplied product and service reviews. Its core business is to provide a forum for people to exchange product and service reviews. Review Everything’s revenues have been growing at a modest pace for the past several years and it is profitable. However, the company has many competitors that release innovative features with alarming frequency. Review Everything really needs a new, innovative service offering to leapfrog the competition.

Review Everything has a dedicated marketing team that constantly monitors the social media space to see how customers perceive its current services. In doing so, the team has learned that many users report spending too much time on the Review Everything site separating “authentic” reviews from “suspicious” reviews. Additionally, many users say that there are so many reviews available for certain products (for example, a DVD player) or services (for example, the Chinese restaurant on Main Street) that they find it difficult to wade through them to get an accurate overall picture.

This market intelligence leads to the idea for SR4U, a revolutionary way to identify, filter, and display online reviews that includes a trainable search agent. Marketing believes this idea could be the innovative service offering that Review Everything has been seeking. Marketing writes a one-page description of SR4U that includes its high-level target features, target customers, and key advantages. The team then sends this description to the New Product Approval Committee, which reviews it at its regularly scheduled Idea Review Meeting (held the second Wednesday of every month).

Senior management (which makes up the New Product Approval Committee) agrees that SR4U represents a significant opportunity to differentiate Review Everything, Inc., in the marketplace. The committee then designates Roger, a business representative from strategic marketing, as the product owner for SR4U.

Management has authorized two weeks to complete envisioning, at which time members of the approval committee will review the envisioning results and make a go/no-go decision to fund the initial development of SR4U. In addition to Roger, management has authorized two filtering subject matter experts (SMEs), a market researcher, and a number of stakeholders to participate in the envisioning. However, they have not authorized the larger expenditure of the full Scrum team during envisioning.

Roger is being asked to use the resources available to him to produce the following:

- Initial product vision, product backlog, and product roadmap
- Validation of the primary assumption that users significantly prefer SR4U-filtered results to unfiltered results (Later in the chapter I will describe how Roger and his colleagues will provide this validated learning.)
- A description of the other important assumptions (hypotheses) about the potential users and feature set that the first product release is supposed to test
- The few key, actionable measures used to test the other assumptions and to learn whether the initial release of SR4U is meeting expectations
- List of questions (known unknowns) that need to be addressed

Without this information, senior management would not have sufficient confidence to make an informed decision as to whether or not to move forward with the initial development.

Visioning

The first thing Roger and the stakeholders do is to create a shared, compelling vision for SR4U. In Scrum, a vision is not an elaborate, several-hundred-page document. If we need this much space to describe our vision, we probably don't understand it. Visions, even of complex products, should be simple to state and should provide a coherent direction to the people who are asked to realize them. Take, for example, President Kennedy's vision to go to the moon: "I believe that this nation should commit itself to achieving the goal, before this decade is out, of landing a man on the

Moon and returning him safely to the Earth” (Kennedy 1961). In 31 words Kennedy was able to express an aggressive, unambiguous vision that, to be realized, would eventually require the efforts of thousands of collaborating people building many complex systems with hundreds of thousands of interrelated components.

When developing products or services, the vision is frequently expressed in terms of how the stakeholders get value. Examples might include one or more areas of value from the categories shown in Figure 17.3.

The format of the vision itself can be anything from a Kennedy-esque statement to a fictitious magazine review. Examples of some popular product or service vision formats are described in Table 17.1 (based in part on Highsmith 2009). You should choose whatever format best suits your organization, envisioning group, and idea.

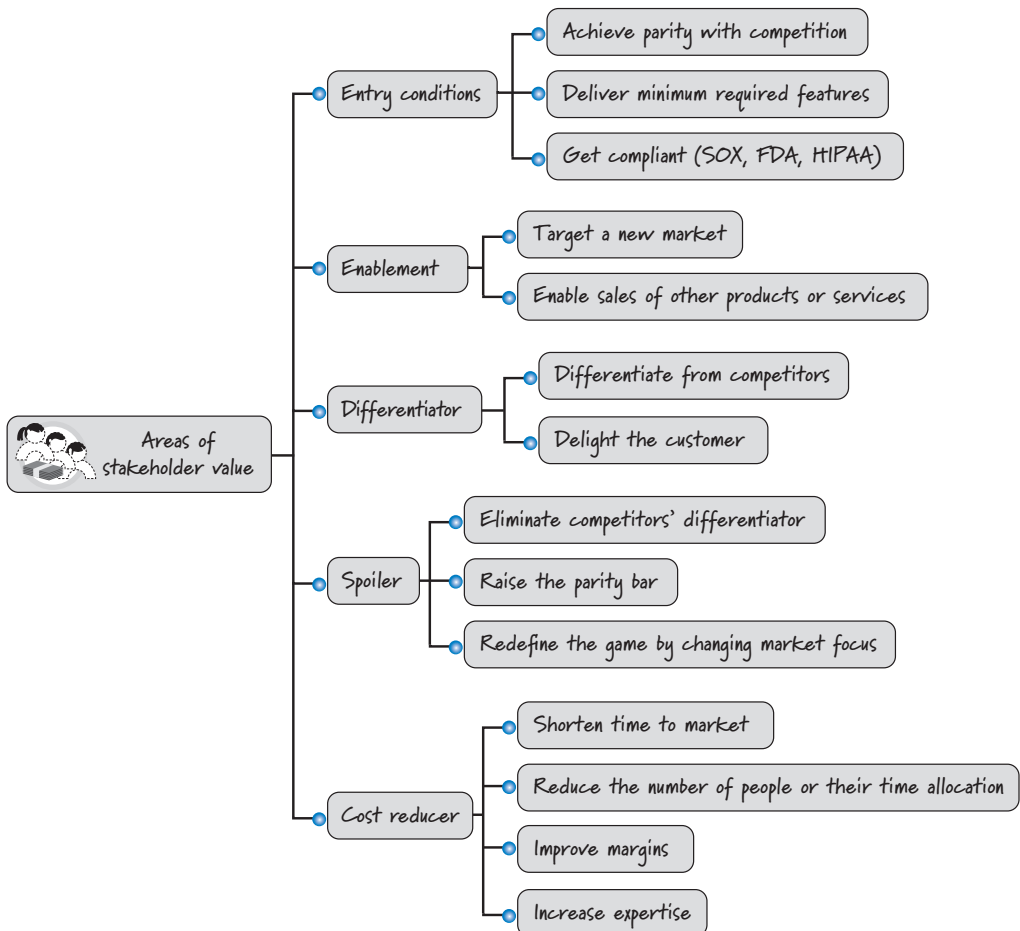


FIGURE 17.3 Areas of stakeholder value

TABLE 17.1 Popular Vision Formats

Format	Description
Elevator statement	Write a 30-second to one-minute quick pitch of the product vision. Imagine you have stepped into an elevator with a venture capitalist and you have to pitch him on your product vision. Could you do it in a short elevator ride?
Product datasheet	Write the product datasheet on the first day. Try to fit it on the front side of a one-page marketing piece.
Product vision box	Draw the box in which you want to put the product when it ships. Can you come up with three or four salient points to illustrate on the box? (Drafting 15 points is easier than drafting three or four.)
User conference slides	Create the two or three presentation slides that you would use to introduce the product at your user conference (or equivalent). Try to avoid any bullet points on your slides.
Press release	Write the press release you want to issue when the product becomes available. Good press releases clearly communicate what is newsworthy in one page or less.
Magazine review	Draft a fictitious magazine review bylined by the solution reviewer in your industry's most popular trade magazine.

At Review Everything, Roger and the stakeholders choose to use the press release format to describe SR4U. They begin by identifying several areas of **stakeholder value** (from Figure 17.3) that SR4U should deliver. The relevant areas are described in Table 17.2.

TABLE 17.2 SmartReview4You Potential Areas of Stakeholder Value

Area	Description
Cost reducer/time savings	SR4U must save its users considerable time when searching for reviews.
Differentiator/delight the customer	SR4U must provide a “wow” experience for its users. Users must feel that the service performed an impressive task for them, helping them make an informed purchase.
Spoiler/raise the parity bar	SR4U should create substantial chaos for its competitors. Their current solutions should immediately look antiquated by comparison. SR4U will establish a new baseline for online review services that others will have to scramble to meet.

Based on these areas of stakeholder value, Roger and the stakeholders craft the following press release (vision statement):

Review Everything, Inc., announced today the successful launch of its new SmartReview4You service. This service provides all online users with their own trainable agent to scour the Internet and identify unbiased, relevant product or service review information.

Remarked Doris Johnson, an avid user of online reviews, “I now have my very own personal assistant that mimics how I find and filter online reviews. It’s amazing—I teach it what I like and don’t like about reviews, then SmartReview4U tears across the Internet finding product or service reviews and automatically weeds out the biased or bogus ones. It does at lightning speed what used to take me forever. This service is a huge timesaver!”

C. J. Rollins, CEO of Review Everything, Inc., said, “We are pleased to offer the world’s first truly smart review service. Since the inception of the Internet people have leveraged the wisdom of the online crowd. However, the crowd can get very noisy at times and it is hard to separate the wheat from the chaff. Our super-smart service does the laborious work of sifting through the huge volume of online review information, eliminating suspicious reviews and returning only relevant ones. You read only the reviews you’d choose to consider if you spent hours searching on your own.”

The new SmartReview4You is available free of charge at the following website: www.smartreview4you.com.

High-Level Product Backlog Creation

Once we have a vision, we are ready to create high-level product backlog items. Although there are many ways to represent product backlog items, I like to use user stories (discussed in detail in Chapter 5). In the terminology of user stories, during envisioning I want to create epics—really large user stories that are consistent with the product level of planning. These epic-level stories align with the vision and provide the next level of product detail for senior management and the Scrum team.

The people who write these stories are usually the same people who created the vision—the product owner, stakeholders, and preferably the ScrumMaster and development team. As a general rule, I want all of my Scrum team members involved in writing these stories. However, as I mentioned previously, if the product development hasn’t yet been approved/funded, the full Scrum team might not be available during envisioning. In those cases, the product owner might want to call in a favor and ask a few technical people with an interest in the product area to help out with story writing.

At Review Everything, SR4U has not yet been approved, so a development team has not been assigned. Therefore, Roger and the stakeholders ask Yvette, an experienced architect, to join them in their story brainstorming session. During the session they create a set of initial epics, including the following:

As a typical user I want to teach SR4U what types of reviews to discard so that SR4U will know what characteristics to use when discarding reviews on my behalf.

As a typical user I want a simple, Google-like interface for requesting a review search so that I don't have to spend much time describing what I want.

As a typical user I want to have SR4U monitor the Internet for new reviews on products or services of interest and automatically filter and report them to me so that I don't have to keep asking SR4U to do it for me.

As a sophisticated user I want to tell SR4U which sources to use when searching on my behalf so that I don't get back reviews from sites I don't like or trust.

As a product vendor I want to be able to show an SR4U-branded review summary for my product on my website so that people can immediately see what the marketplace thinks of my product as determined by a trusted source like SR4U.

Product Roadmap Definition

Once we have the initial vision and a high-level product backlog, we can define our initial product roadmap, a series of releases for achieving some or all of our product vision. When using Scrum, we always develop incrementally. We also try to deploy incrementally when that approach is sensible, meaning we focus on smaller, more frequent releases where we deliver some of the solution before we deliver all of the solution. A product roadmap is an initial overview of these incremental deployments. Of course, if we are planning only a single small release, we don't need a product roadmap.

Releasing frequently doesn't mean we set overly aggressive deadlines; such deadlines frequently result in missed dates. Instead, we focus each release on a small set of **minimum releasable features** (MRFs) around which the stakeholder community shares a strong group consensus. MRFs represent the smallest set of "must-have" features—the features that simply have to be in the release if we are to meet customer value and quality expectations. Some people refer to this set of features as the **minimum viable product** (MVP) or **minimum marketable features** (MMFs). While we might choose to deliver more than the MRFs in a given release, customers would not

perceive enough value if we delivered any fewer. Therefore, it is always important to define the minimum set.

To complement the MRFs, some organizations use the strategy of fixed, periodic releases—for example, a release every quarter—to simplify the product roadmap (see Figure 17.4).

This approach has several advantages. First, it is easy to understand and provides everyone involved (internally and externally) with predictable releases. It also establishes a rhythm, or cadence, to releasing that helps marshal resources in a predictable way and allows for disparate groups to synchronize their plans.

If we use this strategy, we still determine the MRFs for each release. If the MRFs require less time to develop than the fixed time for the release, some additional high-value features will be created. Fixed, periodic releases might not always be applicable if external events (like a conference or a fixed launch date of a co-branded product) are driving the releases, but its benefits make it worth considering.

Each release on the roadmap should have a clearly defined **release goal** that communicates the purpose and desired outcome of the release. A release goal is created by considering many factors, including the target customers, high-level architectural issues, significant marketplace events, and so on.

When creating a product roadmap, we should consider the customers and how they might be segmented into different markets. The roadmap should express how and when to address these different customer segments. In the case of SR4U, the initial customer market is the individual consumer interested in reading helpful reviews before buying a product or service. The SR4U envisioning team further subdivides this market segment into “typical user” and “sophisticated user,” those who want fine-grained control over how SR4U works. The team decides that the initial target will be the typical user.

The SR4U envisioning team can also foresee a future customer base of product and service vendors who would use SR4U to provide an unbiased Internet-wide review history of their offerings on their own websites. However, before vendors will see enough value to pay for the service and the brand, Review Everything, Inc., will first need to establish SR4U as a trusted brand for review aggregation and filtering.

When making a product roadmap, we also should consider high-level architectural or technology issues. For example, on SR4U the principal technology issue is to determine which forms of service access to provide. The team decides to initially provide access via a web browser. However, longer term, it can also envision mobile-device-specific applications for the iOS devices, Android devices, and potentially

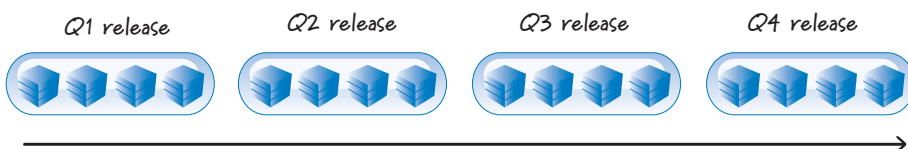


FIGURE 17.4 Fixed, periodic releases

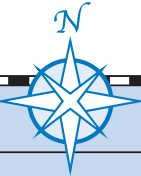
other devices that are tailored to access the SR4U service. Even further down the road, the team also intends to provide an open API that SR4U's partners can access.

When defining a product roadmap, we also might need to allow for any significant market events that could influence the timing of our feature deliveries. Review Everything, for example, always attends the annual Social Media Expo conference. Roger and the stakeholders agree that having a release available by this year's conference (about three months away) would be a great place to get feedback on the service.

Our goal when creating the product roadmap is to consider any factors we deem relevant to help us define a target set of releases for our solution. Remember, though, that this roadmap is simply a rough first approximation of one or a few near-term releases. We must have the right to update the roadmap as better information becomes available.

We must also consider how far into the future our product roadmap will extend. Although our vision might be large and bold enough to require many years to fully realize, it is unlikely that we would attempt to produce a detailed roadmap that would extend completely across such a vision. When using Scrum, we produce the product roadmap as far into the future as is reasonable and desirable. How far into the future your roadmap should extend will depend on your particular circumstances. At a minimum, your roadmap will probably need to cover at least the span of time you are asking people to fund.

Roger and the SR4U stakeholders believe their vision will probably take several years to fully realize, but Roger decides that it would not be practical to try to extend the roadmap out that far given their low level of validated learning and how quickly things change in the online reviews marketplace. Roger and the stakeholders settle on a simple nine-month product roadmap, as shown in Figure 17.5.



	Q3—Year 1	Q4—Year 1	Q1—Year 2
Market map	Initial launch	Better results More platforms	Sophisticated users
Feature/benefit map	Basic learning Basic filtering	Improved learning Complex queries	Define sources Learn by example
Architecture map	100K concurrent web users	iOS and Android	Web services interface
Market events	Social Media Expo	Review Everything User Conference	
Release schedule	1.0	2.0	3.0

FIGURE 17.5 SmartReview4You product roadmap

Other Activities

Envisioning can include any other type of work that those involved agree is relevant to achieving the target confidence threshold. Perhaps we want to do minimal market research into the target customers or users. Or maybe we want to do a quick competitive analysis of the proposed product against other offerings in the marketplace. Or perhaps we want to create a rough business model to help us decide whether the product passes the organization’s “economic filter.”

Some organizations might even decide to organize the envisioning work into one or more sprints. In these cases the assigned team (the envisioning Scrum team if you will) maintains a backlog of envisioning-related work that is prioritized and worked on in short-duration sprint cycles (perhaps one-week sprints). Some of these sprints might involve knowledge-acquisition work, as described in Chapter 5. Examples of knowledge-acquisition sprints might include creating a prototype or proof of concept of the product look and feel, or a critical architectural feature.

For SR4U, Roger and his team (including the SMEs) decide to perform one knowledge-acquisition sprint during envisioning. Before investing in the development of an automated system, Roger first wants to run a simple comparison test to confirm the core assumption that SR4U-filtered reviews are really much more helpful to users than unfiltered reviews (see Figure 17.6).

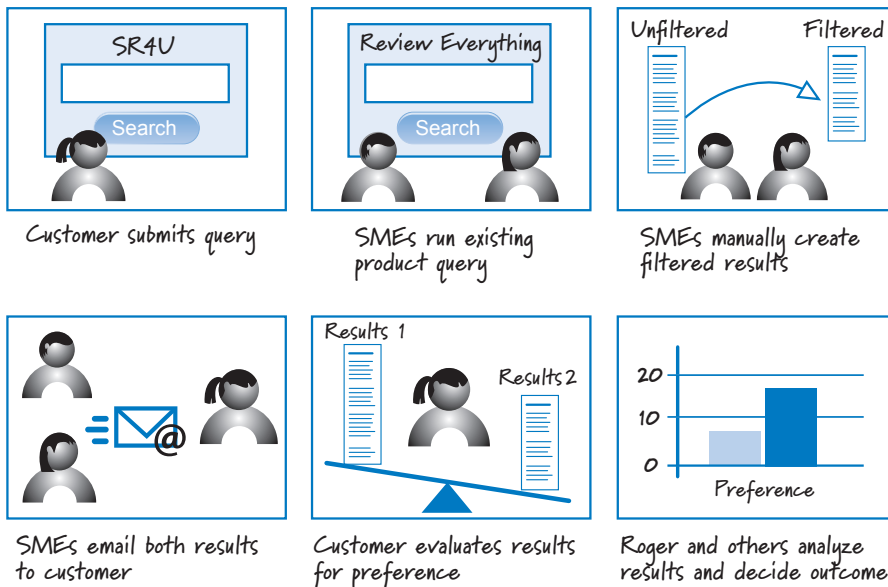


FIGURE 17.6 SR4U knowledge-acquisition sprint storyboard

During the envisioning sprint, the team will mock up one web page (an HTML, Google-simple search page for SR4U) where a small, sample group of users can submit a query for a product or service of their choosing and get back two sets of results. The first set will be the unfiltered reviews that would normally be returned to the query. The second set will be filtered to remove “suspect reviews.” The users will not be told which reviews are filtered and which are not.

The sample users are told ahead of time that their query results will be ready the next day (via email) because, unbeknownst to them, Roger has no intention at this time of developing the technology necessary to automate the generation of their filtered query results. Instead, he asks a couple of SMEs to manually do the filtering and provide the users with both the filtered and unfiltered results. Roger and his team will then interview all members of the sample user group to see which results they prefer and why.

The goal behind this early test is to get basic validation of the core value proposition underlying SR4U—that users will be delighted by the SR4U-filtered set of reviews. If the SMEs can’t manually generate compelling filtered results, Review Everything’s ability to create an expert-system-type product that will deliver value in the marketplace is put in serious doubt.

Senior management also asks Roger to describe the other core assumptions/hypotheses that are not yet validated about the potential users and feature set along with key measures for testing these assumptions. He will collaborate with product marketing people and others to complete this work. Instead of doing an extensive, time-consuming market research study, Roger is planning to use the development of the first release as an experimental tool for discovering what people actually think of SR4U and what they really want in terms of features.

Economically Sensible Envisioning

Envisioning needs to be carried out in an economically sensible way. It should be viewed as an investment in acquiring the information necessary for management to make an informed decision about whether to fund the work required for developing a product based on the idea. If we do too little envisioning, we might find ourselves unprepared to do the first customer-value-creation sprint. On the other hand, doing too much envisioning will create a large inventory of product-planning artifacts that may have to be reworked or discarded when we start to acquire validated learning.

In many organizations, envisioning-type work goes by the name of **project chartering**, **project inception**, or **project initiation**. In some organizations the chartering process is part of a comprehensive stage-gate governance model. Frequently, in this context, chartering is a heavyweight, ceremonial, plan-intensive process based on a set of predicted data. This detailed but not-yet-validated data forms uncertain plans that provide only the illusion of certainty when making a go/no-go funding decision.

In addition, a heavyweight upstream approach is poorly aligned with the agile downstream Scrum development process. This impedance mismatch is like saying, “You can develop using Scrum, but before we approve development we will still need the same artifacts we always require: extensive up-front requirements, a full budget, and a precise schedule.” With this type of misalignment, it will be difficult for an organization to achieve the long-term, high-value benefits from using Scrum.

In Scrum, we keep envisioning as simple as possible. We do just enough up-front predictive planning and knowledge-acquisition work based on the product’s nature, size, and risk level. We allow the details of some other artifacts to be created in a just-in-time fashion. Our goal is to make the best decision we can today using reasonable information obtained in a financially and time-sensitive manner. We acknowledge that what we think we know about the product can and will change once we actually build something and start to subject it to customer and user scrutiny.

I have found several guidelines to be helpful for envisioning in an economically sensible way (see Figure 17.7).

Target a Realistic Confidence Threshold

The confidence threshold defines the minimum level and type of information that is being requested by decision makers to give them enough confidence to make the next-level go/no-go funding decision. Think of the confidence threshold as the bar that must be cleared before we can exit envisioning and subject the product to the scrutiny of portfolio planning—where we apply the economic filter to the product to determine if it meets the organization’s funding criteria. And, if it does, we can get on to the business of validating key assumptions and building the product.

The height of that bar has real economic consequences (see Figure 17.8).

The higher the bar, the more time we need to clear it. Additional time spent during envisioning will likely delay when the product will ship, and that delay has a

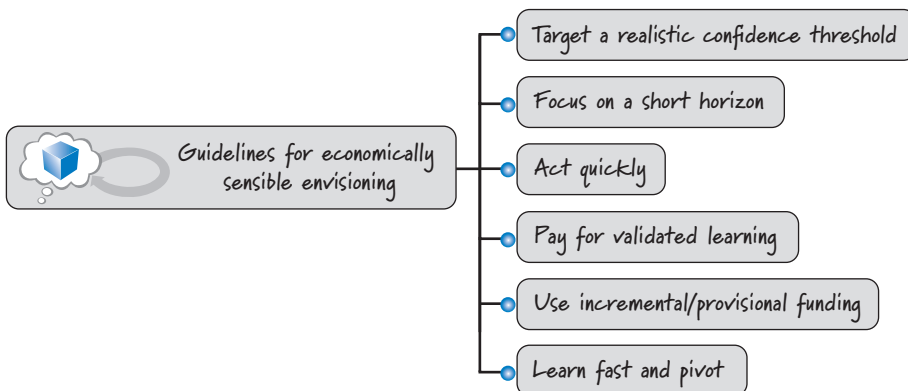


FIGURE 17.7 Guidelines for economically sensible envisioning

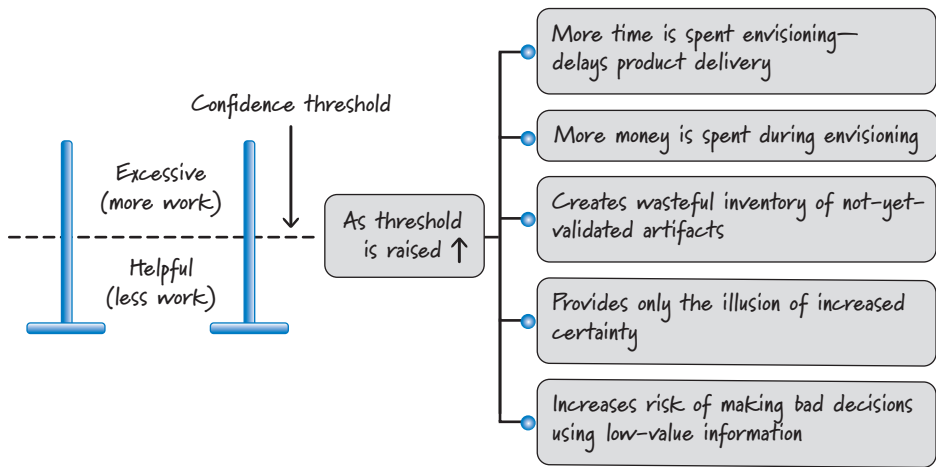


FIGURE 17.8 Consequences of setting the confidence threshold bar too high

cost (see Chapter 3). Envisioning time also has to be paid for, so the higher the bar, the greater the cost of clearing it. More predictive work also creates additional WIP (inventory) that might easily be wasted when things change. And most of that WIP is not yet validated (for example, planning artifacts that predict what might occur in the future), so additional increases to the bar height don't add any certainty to our efforts. Finally, more work can actually increase our risk of making a bad decision to proceed because of the illusion of certainty established by the ever-increasing set of planning artifacts that get produced. More planning artifacts do not imply more certainty or a better funding decision.

As I mentioned in Chapter 14, up-front planning should be helpful without being excessive, so we need to set the threshold to the helpful level, not to the excessive level. What exactly constitutes a helpful level versus an excessive level is organization and product specific. Some organizations are comfortable making decisions under very uncertain conditions while others require a high degree of certainty before proceeding. As the need for certainty increases, so does the effort required to collect data and generate validated learning. There is a practical limit to how much validated learning we can create until we get into development, start building something, and actually validate it with our users. So be realistic about how high the threshold is set.

Also, the threshold for envisioning the next release of a long-lived, core system will likely be less than the threshold for envisioning a new, highly innovative, and potentially expensive product.

Review Everything, Inc., has moved away from heavyweight up-front product-level planning. The approval committee has agreed that the confidence threshold should be set “good enough” or “barely sufficient” to proceed to initial development, where the company can validate assumptions with users. The approval committee is

not looking for a full project plan down to the level of the tasks each person will work on and when. Instead, it wants good clarity on what the goal is for the next set of development work and how Roger plans to measure the results so that he can decide on the next, best course of action.

Focus on a Short Horizon

Don't try to envision too much at one time. Focus primarily on the must-have features of the first candidate release. If we plan on a very broad horizon, chances are we are wasting our time planning for things that might never happen. Plus, if we are developing a new, innovative product, most of our assumptions are not yet validated, so it is very likely that when we subject our product to the uncertain customer environment, we will learn something important that will motivate us to adapt our vision and the plans of what we are building.

For SR4U, Roger's high-level roadmap goes out nine months, but it is really the first release that is the focus. Everyone involved knows that until they actually have a review service that customers can use and comment on, they are all just guessing about the proper feature set. So trying to envision too far into the future will require them to base assumptions upon yet more assumptions, violating the principle of having fewer, short-lived important assumptions.

Act Quickly

Envisioning should not be a long, drawn-out process. It should be fast and efficient. The more quickly we finish, the sooner we get to building something tangible that we can use to validate whether our understandings and assumptions are correct or not.

Time spent envisioning should be included in the calculation of the time required to deliver a solution. The market clock starts ticking the moment the business opportunity becomes known (when the idea is generated) and doesn't stop ticking until we deliver the product. An unnecessarily long envisioning activity will delay product delivery, and that delay cost might be quite expensive. The economics of acting quickly during envisioning are compelling—as Smith and Reinertsen remark, it is the “bargain basement” of cycle-time reduction opportunities (Smith and Reinertsen 1998).

Acting quickly also promotes a sense of urgency to make a product decision. This urgency helps ensure that the proper resources are identified and committed to complete the envisioning work in a timely way.

One way to promote quick movement is to provide an expected completion date to the envisioning team (one of the inputs to envisioning). Not every idea will require the same amount of time to envision. As I mentioned earlier, a new, innovative product idea might require more time to envision than an enhancement or update to a long-existing product. In either case, however, we still want to place reasonable boundaries on the envisioning work so that we can quickly get to the point of validating our assumptions through real feedback.

In the case of SR4U, Roger and the others have two weeks to complete the visioning work. Roger will need to be dedicated full-time to meet this deadline. The filtering SMEs will need to be dedicated half-time during the second week, when they perform the knowledge-acquisition sprint. The market research person will be needed for two days during the first week.

Pay for Validated Learning

Evaluate envisioning activities from an economic perspective based on how they contribute to the acquisition of validated learning regarding the target customer, the target set of features, or the solution. Be wary about performing predictive activities that generate information with a high degree of uncertainty—information that is believed to be valid but has not yet truly been validated with customers or users. These activities purchase low-value information and are not only a bad return on investment; they are also potentially quite wasteful if once we get validated learning we end up discarding or reworking highly uncertain information that is wrong.

Also, generating a lot of low-value, highly uncertain information can clutter our judgment and cause us to believe we understand our situation better than we really do. As a result, we make important decisions under the illusion of certainty (see Figure 17.9).

In the case of SR4U, the content of the product backlog and the product roadmap represents uncertain information. Roger believes that what he has produced

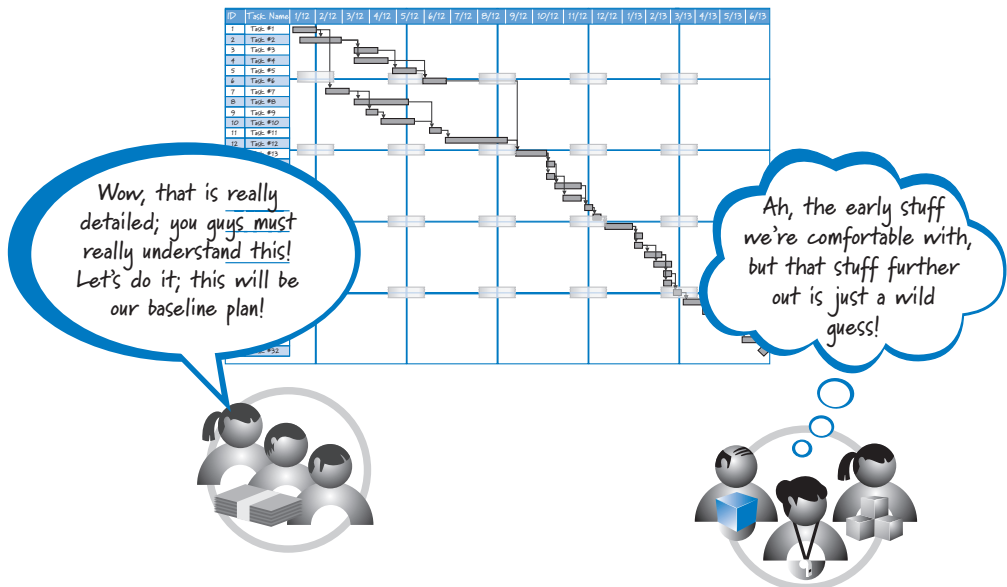


FIGURE 17.9 Decision making under the illusion of certainty

represents a good guess as to what users will want and roughly when they will get it. However, the contents of both are subject to change as the team acquires validated learning during development, so he wants to be careful about how much detail he generates at this time.

During SR4U envisioning, senior management is willing to pay for validated learning regarding the core assumption that users prefer filtered versus unfiltered results. They believe it is economically sensible to buy this information during envisioning before they invest substantially more assets to acquire this same information later. It would be far less sensible to spend considerable money to build the first version of SR4U and then find out that users have no strong preference for filtered versus unfiltered results.

Use Incremental/Provisional Funding

Always consider an incremental or provisional approach to funding product development (see Figure 17.10).

Funding decisions are (or at least should be) constantly being made and remade as better information becomes available. On our first pass through envisioning we shouldn't try to generate enough information to approve and fund all future development of the product, but instead enough information to fund sufficient development to acquire the next important, critical real-world knowledge or feedback regarding our customers, features, or solution approach.

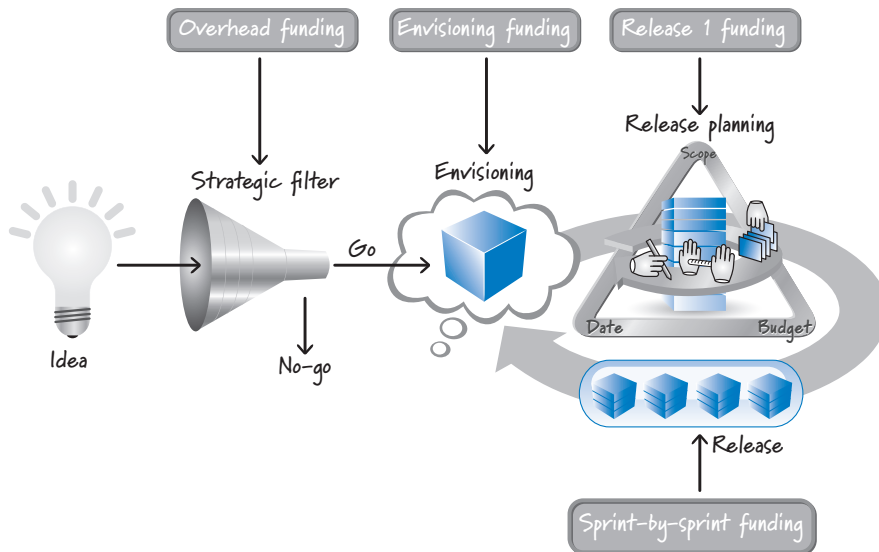


FIGURE 17.10 Incremental/provisional funding

Using **incremental funding**, we would fund just that first small part of the development effort and revisit the funding decision after we have the critical validated learning we paid to get. When we fund incrementally, we can reduce the scope of envisioning and the time it takes to complete it.

Also, remember that the fact that we allocate the funding doesn't necessarily mean we're going to spend the money. As we start getting feedback on a sprint-by-sprint basis, we could choose to pivot to a new vision or simply terminate the product development effort (see Chapter 16 for more details).

In the case of SR4U, Review Everything's policy is that funding is fluid and never given in large chunks but rather in sufficient quantity to validate the next important assumption. Based on feedback and learning, senior management might well continue to spend money that has already been allocated, allocate additional money, or stop funding any future development.

Learn Fast and Pivot (aka Fail Fast)

Envisioning is part of a learn-fast-and-pivot cycle. This approach is sometimes referred to by the catchy alliteration **fail fast**. Simply put, we responsibly and efficiently manage our resources to quickly and cheaply perform envisioning. Then we quickly and cheaply validate our customer, feature, and solution knowledge and assumptions to see whether our vision and product plans meet our business expectations. If we learn that they don't, we pivot quickly and reenvision a more appropriate version of the product, or we simply kill the product and halt any further expenditure.

We can substantially reduce our financial exposure if we are willing to make an informed decision based on reasonable information and then either change directions or terminate the product if we determine that our product vision is incorrect. It is usually far less expensive to start fast and learn fast that we were wrong than to spend a substantial amount of time and money up front to ensure that we make the "right" decision, only to find out eventually that we were wrong. This failing-fast approach is possible only if we are willing to kill a product once we have started spending money to develop it (see the discussion of marginal economics in Chapter 16).

In the case of SR4U, the goal is to very quickly (and cost-effectively) get feedback by getting an initial version of the review learning and filtering capabilities up and running so that people can start using the service. If the team learns after early feedback that the filtered results are not considered by the target user base to be substantially better than the unfiltered results, the company might invest more time in trying to improve the filtering algorithm. However, if after reasonable assets have been invested the company still isn't able to obtain a measurably better set of filtered results, it might be time to pivot and either terminate this product or consider a different direction in which to proceed that leverages the learning to date.

Closing

In this chapter I provided a detailed description of envisioning (product-level planning). I illustrated an approach to envisioning by showing how a fictitious company might create a product vision, high-level product backlog, and product roadmap for its SmartReview4You service. I also illustrated how a knowledge-acquisition sprint during envisioning can be useful in achieving the confidence threshold for completing envisioning. Then I provided guidance on how to perform economically sensible envisioning so that we can better align the up-front product-planning work with the Scrum customer-value-creation work that will follow.

In Chapter 18 I will discuss how we take the outputs of envisioning and use them during release planning.